

Course Overview

ACTL3143 & ACTL5111 Deep Learning for Actuaries
Patrick Laub





Warning

This page is currently being updated for 2024, some details may change.

Your lecturer



Dr Patrick Laub (LIC)

- Bachelor in Software Engineering / Mathematics (Uni. Queensland)
- PhD in Applied Probability (Denmark & Uni. Queensland)
- Post-doc in Lyon, France
- Post-doc in Melbourne, Australia
- Lecturer at UNSW since Jan. 2022

Course objectives

Artificial intelligence and *deep learning* for actuaries.

You will:

- understand common neural network architectures,
- create deep learning models (in Keras) to solve actuarial data science problems,
- gain experience with practical computational tools (e.g. Python).



Moodle & Ed Forum

The Moodle page contains:

- assessment (upload StoryWall, project & exam here),
- lecture recordings,
- link to lecture materials (<https://laub.au/ai>),
- link to Ed forum.

Ed forum will be used for announcements and for questions about the course.

If it is something confidential, then email me.



Lecture plans

1. Artificial Intelligence & Python
2. Deep Learning with Tabular Data
3. Computer Vision
4. Natural Language Processing
5. Recurrent Neural Networks
6. *Away for flexibility week*
7. Distributional Forecasting
8. Interpretability
9. Generative Networks
10. Next Steps



Contact hours

The lectures are 2 hours each week.

The tutorials are a mix of practical coding and theoretical questions.

Make sure to use this time to ask your tutor for guidance on your project.

In later weeks, the tutorials will just be for project help.

Consultation hours will be online and scheduled weekly.



Exercises

On the website, I have added longer exercises for you to try.

Try to finish them around the week they are released (previously they were StoryWall questions).

These will be useful practice for the final exam.

Solutions will not be provided.

I encourage you to collaborate on making class solutions for each exercise (e.g. making Ed forum post for each one, shared Dropbox/Colab, so on).

In the final lecture, I will review any class solutions to the exercises (if they exist) and give feedback.



Final exam

The exam is a take-home format, and thus will be open book and open notes.

You'll be given a neural network task (similar to the exercises, shorter than the project), and will work **individually** to complete it.



Assessment

1. StoryWall (30%)
2. Project (40%)
3. Exam (30%)



StoryWall

There are 7 StoryWall tasks, each worth 5% each.

The best 6 of 7 being counted, adding up to 30%.

These are formative assessments, so are marked pass/fail.

They are due on Friday at noon in Weeks 2, 3, 4, 5, 7, 9, 10.

I'll release them all at the start of term; I suggest looking at them 2 weeks before the due date.



A complete deep learning project

Individual project over the term. You will:

- specify a *supervised learning* problem,
- collect and clean the data,
- perform an exploratory data analysis (EDA),
- create a simple (non-deep learning) benchmark model,
- fit two different deep learning architectures,
- perform hyperparameter tuning,
- write a discussion of the results.



Project components

The deliverables for the project will include:

1. Report Part 1 due at noon on Friday in Week 5 (10%),
2. Recorded presentation due at noon on Friday in Week 8 (15%),
3. Report Part 2 due at noon on Monday of Week 10 (15%).



Due dates

All due dates are at noon of the following weeks (“SW” = StoryWall):

- | | |
|--|--|
| 1. None | 6. None |
| 2. SW ₁ (Fri) | 7. SW ₅ (Fri) |
| 3. SW ₂ (Fri) | 8. Presentation (Fri) |
| 4. SW ₃ (Fri) | 9. SW ₆ (Fri) |
| 5. SW ₄ (Fri) and Report I (Fri) | 10. Report II (Mon) and SW ₇ (Fri) |



Late policy

If submitting late, you must apply for special considerations through UNSW central system. If you ask me for an extension, I will refer you to the special considerations system.

Without special consideration, **late StoryWalls will not be marked**. I have noticed that special considerations will not be granted for StoryWall tasks if you can still get full marks without that task.

For the project, the general policy is:

Late submission will incur a penalty of 5% per day or part thereof (including weekends) from the due date and time. An assessment will not be accepted after 5 days (120 hours) of the original deadline unless special consideration has been approved.



Example: Late policy for Report Part 2

Report Part 2 (worth 15% course grade) is due Week 10 Monday noon.

If you submit without special consideration on:

- Week 10 Monday 11:59 am, you have no late penalty.
- Week 10 Monday 12:01 pm, you have a **5% penalty**.
- Week 10 Tuesday 12:01 pm, you have a **10% penalty**.
- Week 10 Wednesday 12:01 pm, you have a **15% penalty**.
- Week 10 Thursday 12:01 pm, you have a **20% penalty**.
- Week 10 Friday 12:01 pm, you have a **25% penalty**.
- Week 10 Saturday 12:01 pm, you will **get 0 marks**.

E.g. a submission on Tuesday 12:01 pm (10% penalty) which was graded as 80/100, would be recorded as 72/100, and hence an overall course grade of 10.8% out of the maximum 15%.



Special case: Late policy for Report Part 1

However, as a special case **just for Project Report Part 1**, I will not apply the 5% per day penalty for the first 72 hours after the deadline.

Report Part 1 is due Week 5 Friday noon.

If you submit without special consideration on:

- Week 6 Monday 11:59 am, you have no late penalty.
- Week 6 Monday 12:01 pm, you have a **20% penalty**.
- Week 6 Tuesday 12:01 pm, you have a **25% penalty**.
- Week 6 Wednesday 12:01 pm, you will **get 0 marks**.



Exam

The exam will test the concepts presented in the lectures. For example, you will be expected to:

- provide definitions for various deep learning terminology,
- suggest neural network designs to solve risk and actuarial problems,
- give advice to mock deep learning engineers whose projects have hit common roadblocks,
- find/explain common bugs in deep learning Python code.



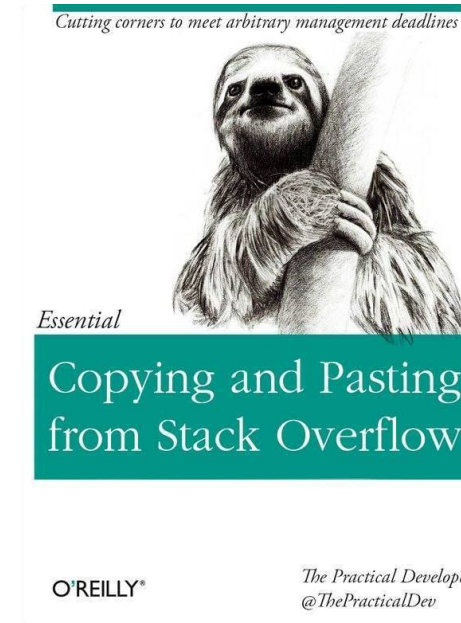
Copying code...

If you copy, tag it:

```
1 # Suppress endless warnings from Keras.
2 # Source: https://stackoverflow.com/a/38645250
3 import tensorflow as tf
4 tf.get_logger().setLevel('INFO')
```

Even if you then edit it a little:

```
1 # Create a basic Convolutional Network.
2 # Adapted from: https://www.tensorflow.org/tutorials/images/
3 model = models.Sequential()
4 model.add(layers.Input((32, 32, 3)))
5 model.add(layers.Conv2D(32, (3, 3), activation='relu'))
6 model.add(layers.MaxPooling2D((2, 2)))
7 model.add(layers.Flatten())
8 model.add(layers.Dense(64, activation='relu'))
9 model.add(layers.Dense(10))
```



Recommended reading.